|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 变量名 | 属性/标记 | 连续/离散 | 类型 | 上下界 | 离散变量个数 |
| Song\_name |  |  | string |  |  |
| Song\_popularity | Label | Continuous | int | (0,100) |  |
| Song\_duration\_ms | Attribute | Continuous | int | (12k,1.8m) |  |
| acousticness | Attribute | Continuous | double | (0,1) |  |
| daceability | Attribute | Continuous | double | (0,0.99) |  |
| energy | Attribute | Continuous | double | (0,1) |  |
| instrumentalness | Attribute | Continuous | double | (0,1) |  |
| key | Attribute | Discrete | int |  | 12 |
| liveness | Attribute | Continuous | double | (0.01,0.99) |  |
| loudness | Attribute | Continuous | double | (-38.8,1.58) |  |
| audiomode | Attribute | Discrete | int |  | 2 |
| speechiness | Attribute | Continuous | double | (0,0.94) |  |
| tempo | Attribute | Continuous | double | (0,242) |  |
| timesignature | Attribute | Discrete | int |  | 6 |
| audiovalence | Attribute | Continuous | double | (0,0.98) |  |

1. 使用Python编程构建线性回归算法，预测音乐流行度。

2. 数据集预处理：

a) 离散属性连续化

b) 连续属性归一化normalization

c) 共线性的检测与处理(关于共线性的检测：以下是一个使用Python语言实现的基于方差膨胀因子（VIF）的共线性检测代码：

import pandas as pd

import numpy as np

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

# 假设数据存在一个名为"df"的DataFrame对象，其中包含四个属性x1、x2、x3和x4。

# X为输入属性，y为输出属性

X = df[['x1', 'x2', 'x3', 'x4']].values

y = df['y'].values

# 计算VIF

vif = pd.DataFrame()

vif["variables"] = ['x1', 'x2', 'x3', 'x4']

vif["VIF"] = [variance\_inflation\_factor(X, i) for i in range(X.shape[1])]

# 输出结果

print(vif)

在该代码中，使用了statsmodels库中的variance\_inflation\_factor函数来计算VIF。如果任何一个属性的VIF值大于5，就可以认为存在共线性问题。去除共线性使用PCA: 以下是一个基于主成分分析（PCA）的共线性处理代码，用于去除共线性：

import pandas as pd

import numpy as np

from sklearn.decomposition import PCA

# 假设数据存在一个名为"df"的DataFrame对象，其中包含四个属性x1、x2、x3和x4。

# X为输入属性，y为输出属性

X = df[['x1', 'x2', 'x3', 'x4']].values

y = df['y'].values

# 使用PCA进行特征降维

pca = PCA(n\_components=X.shape[1])

X\_pca = pca.fit\_transform(X)

# 输出每个主成分的贡献率

print(pca.explained\_variance\_ratio\_)

# 输出每个主成分的权重

print(pca.components\_)

# 输出降维后的属性矩阵

print(X\_pca)

)

3. 使用80%的数据集进行训练，20%的数据集进行测试。需要报告训练误差和测试误差。请注意，需要保证模型能够在测试集上进行准确的预测。

4. 使用可视化技术来展示你的模型和数据集。可以使用Python的Matplotlib库来完成该任务。

5. 报告模型的性能和准确率。可以使用Python的Scikit-Learn库中的mean\_squared\_error函数来计算均方误差（MSE）。如果预测出来的流行度和真实值的差距在1内判定为预测正确，依此计算模型的准确率。

6. 数据在song\_data.csv中，包含18835个样本，每个样本包含13个属性与1个实值标记，属性包括连续数据属性和离散属性，你要根据属性来预测标记（也就是流行度song\_popularity）。以下表格是对于变量的描述

\begin{table}

\centering

\begin{tblr}{

}

变量名 & 属性/标记 & 连续/离散 & 类型 & 上下界 & 离散变量个数 \\

song\\_name & ~ & ~ & string & ~ & ~ \\

song\\_popularity & Label & Continuous & int & (0,100) & ~ \\

song\\_duration\\_ms & Attribute & Continuous & int & (12k,1.8m) & ~ \\

acousticness & Attribute & Continuous & double & (0,1) & ~ \\

daceability & Attribute & Continuous & double & (0,0.99) & ~ \\

energy & Attribute & Continuous & double & (0,1) & ~ \\

instrumentalness & Attribute & Continuous & double & (0,1) & ~ \\

key & Attribute & Discrete & int & ~ & 12 \\

liveness & Attribute & Continuous & double & (0.01,0.99) & ~ \\

loudness & Attribute & Continuous & double & (-38.8,1.58) & ~ \\

audio\\_mode & Attribute & Discrete & int & ~ & 2 \\

speechiness & Attribute & Continuous & double & (0,0.94) & ~ \\

tempo & Attribute & Continuous & double & (0,242) & ~ \\

time\\_signature & Attribute & Discrete & int & ~ & 6 \\

audio\\_valence & Attribute & Continuous & double & (0,0.98) & ~

\end{tblr}

\end{table}